Tenable Security Center Interface

Michael Wornow

Mentors: John Donaldson[1] and Matthew Myrick[1]

[1]Lawrence Livermore National Laboratory

Introduction

Tenable's Security Center is a standalone commercial product that regularly scans network assets for vulnerabilities and malware. It comes with a robust set of features that enable the data it compiles to be filtered, sorted, and displayed in a variety of ways.

Problem

Despite it's capabilities, the **Security Center** web application doesn't integrate nicely into the **Secure Operations Center (SOC)** and **Splunk database** that LLNL currently uses to monitor its networks.

Thus, it is often difficult for security analysts to detect patterns between data collected by Security Center and data collected by other network monitoring tools.

Solution

In order to integrate Security Center's valuable network and vulnerability data into the SOC and Splunk, I used **Security Center's API** to pull data from the Security Center database and push that data into a separate interface that could then be fully integrated within LLNL's SOC and Splunk database.

Method

The Security Center API offers a **vuln:query** method that returns a **JSON (J**ava**S**cript **O**bject **N**otation**)** string of data.

The two most relevant datasets for this interface were:

- **SumIP**- a list of IP's and their scan results
- **VulnDetails** – a detailed rundown of the vulnerabilities of a specific IP

I used PHP **cURL requests** to get data from the Security Center database. Then I displayed the results in a paginated, sortable, and searchable table.

Interface Views

1. Index.php: Returns a list of IP addresses with their overall scan results.

Tenable Database Interface | Search for an IP | Search IP

Vulnerable Systems

25 results per page — Search: All for: Type something here...

IP Address	DNS	Common Platform Enumeration (CPE)	Score	Total Vulns	Vulnerabilities Graph (Sort by # of: Info | Low | Medium | High | Critical)
134.9.	llnl.gov	cpe:/o:redhat:enterprise_linux:3:update9:es	571	158	99 22 26 6
128.15.	.llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1:x64-enterprise	355	212	165 6 13 27
128.15.	.llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1	270	244	203 9 17 13
128.15.	.llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1	270	243	202 9 17 13
128.15.	.llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1	247	242	201 9 16 15
128.15.	.llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1	227	238	199 9 16 13
128.15.	.llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1:x64-enterprise	215	197	164 6 13 13
128.15.	llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1	214	206	171 6 16 12
128.15.	llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1	207	187	155 14 12
128.15.	.llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1:x64-enterprise	205	195	163 6 13 12
128.15.	.llnl.gov	cpe:/o:microsoft:windows_server_2008:r2:sp1:x64-enterprise	205	195	163 6 13 12
134.9.	llnl.gov	cpe:/o:microsoft:windows_7::sp1:enterprise	59	136	119 12
134.9.	llnl.gov	cpe:/o:microsoft:windows_server_2008:r2	50	44	39
134.9.	.llnl.gov		42	60	42 6 12
134.9.	.llnl.gov	cpe:/o:microsoft:windows_7:::enterprise	26	81	71 8

2. VulnDetails.php: Provides detailed information on all of the vulnerabilities of a specific IP address. Requires a **GET** parameter called **ip**, which holds the IP address you want to get vulnerability data on.

<< Go back to the Vulnerable Systems Table

Vuln Details for 128.15.

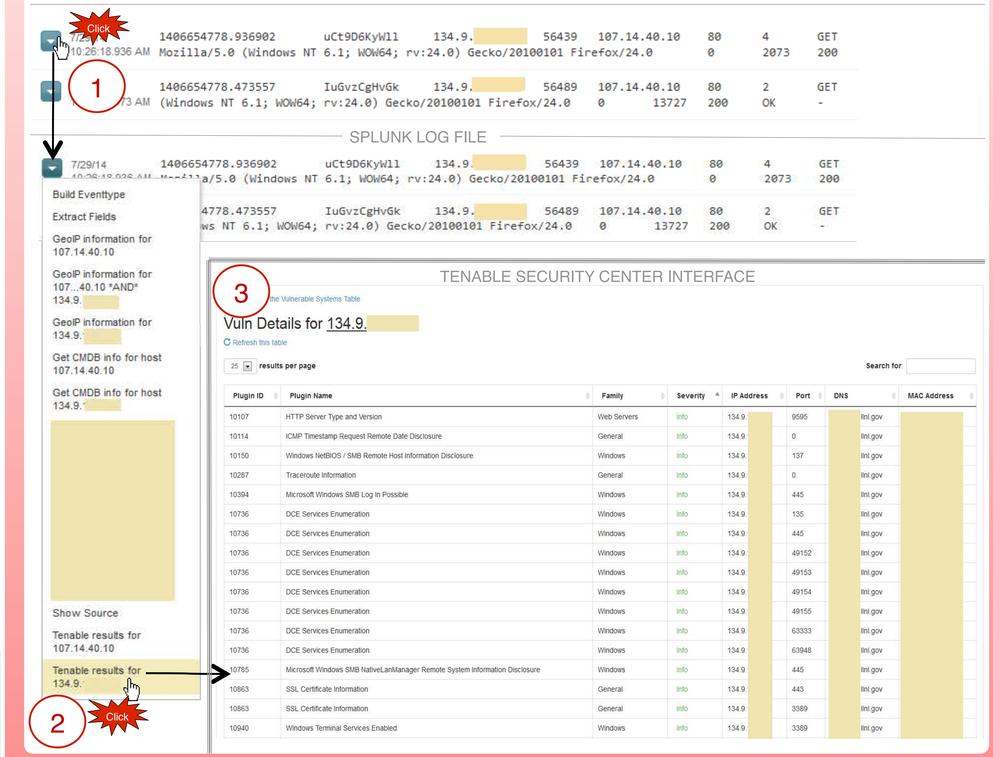
Refresh this table

25 results per page — Search for:

Plugin ID	Plugin Name	Family	Severity	IP Address	Port	DNS	MAC Address
10399	SMB Use Domain SID to Enumerate Users	Windows : User management	Info	128.15.	445		
10860	SMB Use Host SID to Enumerate Local Users	Windows : User management	Info	128.15.	445		
10902	Microsoft Windows 'Administrators' Group User List	Windows : User management	Info	128.15.	0		
10913	Microsoft Windows - Local Users Information : Disabled accounts	Windows : User management	Info	128.15.	0		
10914	Microsoft Windows - Local Users Information : Never changed passwords	Windows : User management	Info	128.15.	0		
10915	Microsoft Windows - Local Users Information : User has never logged on	Windows : User management	Info	128.15.	0		
17651	Microsoft Windows SMB : Obtains the Password Policy	Windows : User management	Info	128.15.	445		
38153	Microsoft Windows Summary of Missing Patches	Windows : Microsoft Bulletins	Info	128.15.	0		
57033	Microsoft Patch Bulletin Feasibility Check	Windows : Microsoft Bulletins	Info	128.15.	445		
71313	MS13-098: Vulnerability in Windows Could Allow Remote Code Execution (2893294)	Windows : Microsoft Bulletins	High	128.15.	445		
71320	MS13-105: Vulnerabilities in Microsoft Exchange Server Could Allow Remote Code Execution (2915705)	Windows : Microsoft Bulletins	Critical	128.15.	445		
72428	MS14-005: Vulnerability in Microsoft XML Core Services Could Allow Information Disclosure (2916036)	Windows : Microsoft Bulletins	Medium	128.15.	445		
72430	MS14-007: Vulnerability in Direct2D Could Allow Remote Code Execution (2912390)	Windows : Microsoft Bulletins	High	128.15.	445		
72432	MS14-009: Vulnerabilities in .NET Framework Could Allow Privilege Escalation (2916607)	Windows : Microsoft Bulletins	High	128.15.	445		

Splunk Integration

A simple link from LLNL's **Splunk** database to **vulndetails.php?ip=*<insert ip here>*** returns a webpage listing all the vulnerabilities for that IP.



Conclusion

The **Tenable Security Center Interface** cleanly integrates with Splunk and the SOC, fetches data from Security Center almost instantly, and displays the data in an easy to read table.

Future Work

The only limitations that I encountered for this project stemmed from the Security Center API itself, for there were inconsistencies in the way that datasets were returned. Sort functionality was largely missing, and several API calls didn't accept start/end offsets. These minor issues required unnecessary data filtering in my PHP files and more GET requests to be sent to Security Center than it should have needed.

In the future, I would like to change how my interface fetches data in order to improve its speed. This would entail **caching data** to avoid repetitive requests, creating **faster algorithms** to filter IPs, and giving the user more **precise options** to specify what data is needed.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

LLNL-POST-658315